Decision Trees for Dynamic Decision Making And System Dynamics Modelling Calibration and Expansion

A Thesis Submitted to the

College of Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the degree of Master of Science

In the Department of Computer Science

University of Saskatchewan

Saskatoon

By

Wenyi An

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

Many practical problems raise the challenge of making decisions over time in the presence of both dynamic complexity and pronounced uncertainty regarding evolution of important factors that affect the dynamics of the system. In this thesis, we provide an end-to-end implementation of an easy-to-use system to confront such challenges. This system gives policy makers a new approach to take complementary advantage of decision analysis techniques and System Dynamics by allowing easy creation, evaluation, and interactive exploration of hybrid models. As an important application of this methodology, we extended a System Dynamic model within the context of West Nile virus transmission in Saskatchewan.

Acknowledgements

I would like to acknowledge and extend my heartfelt gratitude to my supervisors, professor Osgood and professor Grassman, for their generous assistance and invaluable advice. Professor Osgood was always there to provide every kind of help and advice in dealing with the difficulties in life and my studies. He was tireless in his efforts to provide conditions that helped me succeed with my research. Professor Grassman trained me to accurately structure the results of my research. He also meticulously and thoughtfully guided me to effectively organize my ideas. In different ways, each taught me to approach a problem and improved my skills in generating creative ideas, in implementing those ideas, and in articulating them.

Professor Tasha, Dr. Curry, Dr.Chen and Karen Yee, gave me very helpful suggestions. Professor Tasha generously gave me lots of suggestions about my project and modelling. Dr. Curry and Dr. Chen provided me with the values in my model. Karen Yee, one of my best colleagues, encouraged me in my personal life and research. Indeed, I also learned invaluable lessons from her.

To my parents, who offered me unconditional love and support in my life. I want to especially offer my gratitude to my mother, the bravest person I know, who has fought with illness much of her life. She is such an inspiration to me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1. Motivation

This thesis seeks to construct an integrated toolset – accessible to non-technical policy makers – to support adaptive dynamic decision making with respect to control of West Nile virus (a mosquito-borne disease) infection in Saskatchewan, and which supports a broad range of other dynamically complex decision problems. We use this system to demonstrate the use of that approach with a model refined for West Nile virus.

System Dynamics is an approach to enhance learning regarding and management of complex systems [1]. It focuses on internal feedback loops and accumulations that affect the behavior of the whole system [2]. By simulating the evolution of transmission and policy impact, System Dynamics models can play a very important role in disease control and prevention.

A decision tree is a formalism that uses a tree-like graph or model of decisions and uncertainties over time, and their possible consequences. Decision trees are often used in Operations Research and Management Science, specifically in Decision Analysis, to help identify a "decision rule" – a mapping from uncertain situations to decisions most likely to reach the desired goal. Another use of decision trees is as a descriptive means for calculating conditional probabilities. Decision trees support roll-back values (the process of calculating from terminal node to root node of the decision tree) to determine the best decision for each decision node throughout the entire decision tree once the consequences of all scenarios have been calculated. Right now, decision trees are used widely in health, most notably at the clinical level, but also including public health domains such as disease transmission [3].

I improved a model of West Nile virus (WNv) disease that was constructed by Yee[4] representing the diffusion and control of the disease as it is transmitted through mosquitoes in Saskatchewan.

West Nile virus (WNv) is a potentially lethal mosquito-borne pathogen. Upon infection of a person, the pathogen triggers diverse clinical symptoms, like fever accompanied by malaise, lymphadenopathy, etc. Symptoms of infection in humans vary from subclinical illness to death [5].

The virus was first recognized in 1937 from a febrile woman in the West Nile District of Uganda [6], but appears predominantly as a milder illness across the African continent, where it is endemic. WNv arrived in North America in 1999. Following importation of an infected animal to the Bronx Zoo [7] in New York City, the first sign of WNv in North America was a dead American Crow found in New York, which subsequent investigation revealed had died from WNv induced viremia [6]. In following years, the zoonotic pathogen spread rapidly across the United States and Canada. Saskatchewan suffered the highest incidence of WNv in Canada in 2003 and 2007, with the Saskatoon Health Region reporting 6.5% and 25% of the provincial Canadian cases in 2003 and 2007, respectively [8].

In real-world situations, WNv transmission is associated with many uncertainties; the temperature, rainfall and many other factors are governing the size of mosquito populations. Temperature can also affect how quickly the virus matures within mosquitos.

With the development of computer technology, modelling techniques play an increasingly important role in simulating infectious disease transmission, and in helping policy makers choose strategies for preventing and controlling infectious diseases. Several research models have been constructed to investigate the dynamics and control of WNv [9; 10]. However, WNv is distinguished from many other diseases by virtue of presenting a need to make decisions on a week-by-week basis during summer months, due to pronounced uncertainties, particularly with regards to temperature, precipitation, and mosquito populations.

For the case of WNv, we wish to use a System Dynamics model to make a series of policy decisions over time in a situation where the choice of the best decision at a given point depends not only on past decisions but also on the sequence of chance events observed. Many decision makers are facing this kind of problem but there is currently poor support for dealing with it. A hybrid approach was proposed by Osgood and Kaufman [11] that combines System Dynamics and decision analysis method together with policy making. Implementing this hybrid method is central to the system implemented here. This method provides the advantages offered by combining the System Dynamics and decision analysis approaches.

1.2. Epidemiology of West Nile virus

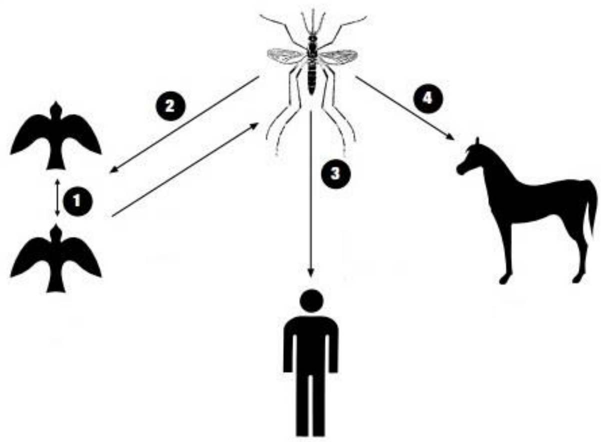

Figure 1.1: Transmission cycle of the West Nile Virus (picture from Pennsylvania State University [12])

The basic life cycle of West Nile virus is depicted in Figure 1.1. (1) Birds serve as reservoirs of infection, and bird-to-bird transmission has been demonstrated in caged birds. (2) The main vector for the spread of WNv are mosquitos. Mosquitos bite birds, and the birds develop high enough level of virus to allow for infecting other mosquitos that bite them. Therefore, there is a WNv transmission cycle between birds and mosquitos. (3) The virus is believed to not reach high enough levels in mammals (including humans) to allow for transmission back to an uninfected mosquito. (4) Some mammals bitten by an infected mosquito may test positive for WNv even though they will not exhibit symptoms [12].

Culex tarsalis (a kind of mosquito) is believed to serve as the primary rural vector of WNv throughout its distribution from Canada to Mexico [13], including within Saskatchewan. Within this thesis, we seek to study the diffusion and control of disease transmitted by this mosquito.

1.3. An overview of hybrid modeling approach

A hybrid modeling approach combining System Dynamics and Decision Trees is used. This approach can provide many benefits for practical decision making when trying to make decisions over time concerning interventions that might have outcomes that are both complex and depend in an important way on various uncertainties that play out over time. This thesis describes a process for linking up a decision tree with a System Dynamics model. In order to do this, a West Nile virus [4] model in Vensim (a modeling software has good performance of data flow transfer between different situation) – significantly extended here – is used as an example. The decision tool could link to any other modeling software. In this thesis, I focused on the models created by Vensim. The use of a System Dynamics model to study the diffusion and control transmission by the mosquito has several advantages [14], reflecting the complex nature of the transmission process, multi-species population dynamics, and complex and time-varying intervention impacts.

The number of cases of WNv in a year in Saskatchewan exhibits orders of magnitude differences year-to-year [8]. The huge differences from year to year, are based on factors such as temperature and (to a lesser degree) rainfall. The best policy in one year (e.g., doing nothing) may be a dangerous policy in another year. Therefore, the people who work in the Ministry of Health and various health regions throughout Saskatchewan have frequent (weekly) meetings throughout the mosquito season to make decisions regarding mosquito control. The combination of dynamic complexity and large uncertainties over time pose a big problem regarding how to best make decision taking into account uncertainties (mosquito population, environmental conditions and human behaviour). In mosquito control – as in many other cases– policymakers need to make complex dynamic choices when they cannot predict the course of the important factors outside their control, but where those factors strongly affect the impact of different possible decisions. That means they need to not only focus on a dynamic model (which can give projections of outcomes of a particular decision under particular scenarios involving fixed assumptions regarding external factors) but they also need to deal with adaptive planning given uncertainty [8]. Our objective in such situations –termed dynamically complex decision problems in [15]– is to avoid a pre-set plan, and instead to adaptively make choices over time [8]. Such "dynamically complex decision problems" problems exhibit some notable features: (1) Stakeholders cannot count on one particular future trajectory unfolding for things outside of their control. (2) The outcomes of a given decision can have very different consequences, depending on how external factors play out. (3) Decisions need to be made over time as policymakers observe how uncertainties resolve [8].

1.4. Thesis contributions

Our contribution in this thesis builds much additional functionality in a small proof-of-concept prototypes constructed by two students (Amy Gao and Tony Leung) [16], extending and implementing software to combine decision analysis and System Dynamics simulation in support of solving dynamic decision problems with decisions and uncertainties over time.

I built a visualization toolkit. It can represent an explored decision tree structure in XML.

A model of West Nile virus (WNv) disease was constructed by Yee [17]. The model has proved to be a useful and convincing general simulation of WNv disease transmission among mosquitoes, birds, and humans. There are three main shortcomings in the model: (1) Certain key processes (e.g., hibernation, egg-laying, bird migration) are not represented (2) the parameters were not yet parameterized or calibrated based on datasets from scientific records and (3) only the effects of relationships among mosquitoes, birds, and humans in a certain area are considered in disease transmission; that is, disease diffusion among populations from different areas, or even between neighboring areas, is not modelled. In this study, we seek to address shortcomings (1), (2) and (3).

1.5. Thesis organization

The remainder of the thesis is organized as follows. Chapter 2 reviews background information on decision analysis and System Dynamics modeling. Chapter 3 presents the tree viewing application used to browse the tree structure and show information on preferred decisions and rollback values. Chapter 4 compares the original System Dynamics model version constructed by Yee[4] and my expanded version. Results are discussed in Chapter 5. Chapter 6 includes a summary, contribution of my research and future work.

Chapter 2

Background and related work

In order to establish better understanding, this chapter will describe the background of my work. Firstly, I will introduce Decision Trees. Secondly, this chapter will represent two kinds of models which are relevant to my work. The end of this chapter briefly explains a hybrid modeling approach combining decision trees and System Dynamics modeling.

2.1. An overview of Decision Tree and terminology

To describe the decision analysis approach, I will use a West Nile Virus Disease Dynamics model as an example.
Figure 2.1 depicts an example decision tree, which by convention is oriented such that time runs from left to right. This tree was constructed by MPH graduate Karen Yee[4]. The elements of the decision tree are associated with their own terminology.

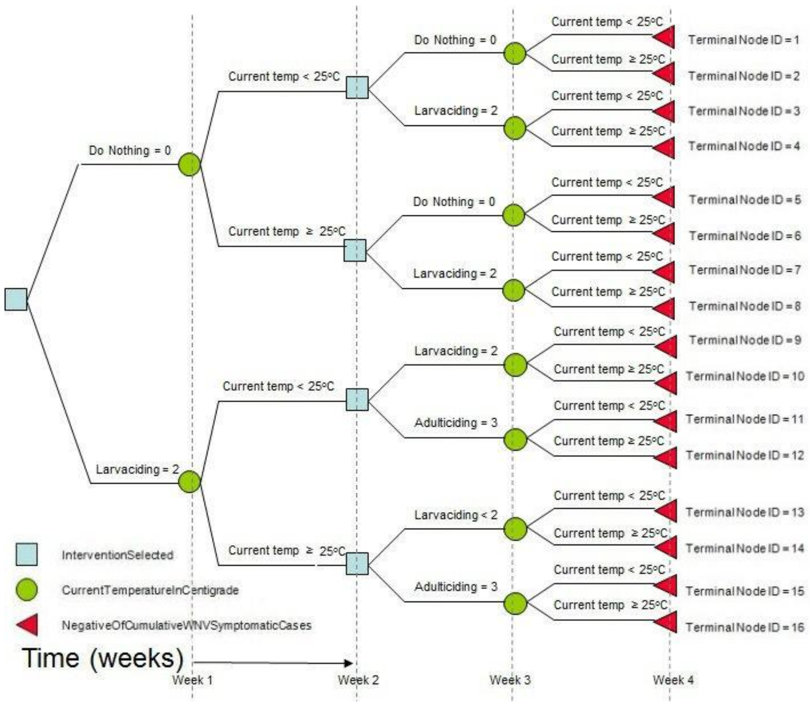

Figure 2.1: Decision Tree created by Graphical User Interface for the West Nile Virus example

The first set of terms relates to the nodes (vertices) seen in the tree:
Decision node: Figure 2.1 presents a decision tree laying out the set of uncertainties, decisions and relationship of this example. The square nodes in Figure 2.1 are called decision nodes. They represent choices to be made at a particular point in time -- here, Adulticiding (insecticide against the adult mosquitoes) and Larvaciding (insecticide against the larval mosquitoes).
Event node: The circle nodes describe uncertainties over time (e.g. the mean temperature over a week being closer to 20 ℃ or 30 ℃).

Terminal nodes: At the end of each path from the root to the leaves of the tree, there are some triangular nodes (leaves) termed “terminal nodes”. Each terminal node is associated with a unique sequence of specific events and particular decisions on the path from the root node to that terminal node. This path is termed a *scenario*. For example, in Figure 2.1, the terminal node which has ID=3 is associated with a path from root node and does nothing during week 1, after that, CurrentTemp is in the area of 20 ℃℃ during week 2. In week 3, larvaciding is initially undertaken, and in the last week (week 4), a lower temperature is again experienced.
Rollback values: Rollback values represent the reward for a scenario, and are calculated recursively. For a terminal node, the rollback value equals the terminal value. For an event node, the rollback value is the expected value of the outcomes of that node – that is, sum of each branch’s probability multiplied by the rollback value of this branch’s child. For a decision node, the rollback value is set equal to the maximum (i.e. most desirable) rollback value of the child nodes– a reflection of the fact that any child can be chosen by an appropriate decision.

2.2. Decision Tree background

The foundations of Decision Analysis were laid in the 1960s by Raiffa and Schlaifer [18] and Howard [19]. Since that time, policy makers and analysts have used decision analysis to conduct analyses to inform decision making in a wide variety of fields. Much work has further taken advantage of the capacity of decision analysis to perform multiple variable analyses to predict, describe or classify an outcome [20]. Decision analysis has been particularly widely applied in the health area, particularly in clinical decision making and cost-effectiveness studies, but has also been applied in epidemiology.

In 2004, a decision tree was employed to decide if it was necessary to implement deceased organ donors West Nile Virus screening in heart transplantations [21]. In 2006 experts evaluated the cost effectiveness of WNv vaccination compared with no vaccination by using decision trees [22]. Such trees have been not only used with respect to financial aspects by processing various governing factors, they also can predict the consequences of certain actions. For example, in the West Nile context, precipitation and temperature can serve as very significant predictors of presence or absence of larvae in the summer; decisions include a variety of policy options (larvaciding, adulticiding, issuing advisories, source reduction) in addition to doing nothing.

2.3. An overview of modeling

2.3.1 Agent-based Modeling with AnyLogic

AnyLogic, a simulation modeling tool can help with multiple types of models. The software supports the most common simulation methodologies: System Dynamics, agent based modeling, and discrete event (process-centric) modeling. AnyLogic allows users to construct declarative graphical representations of a model through a user interface [23]. In an agent-based model, the agents can maintain their own state and execute miscellaneous behaviors [24]. In an Agent-based model, the agents are embedded in an environment (e.g., spatial environment, network, multi-level environment). Each agent is associated with state; state charts are often employed to represent different stages in which the agent could be present. Furthermore, each agent is associated with rules of evolution, and actions by which they can affect the environment. By employing agent-based modeling, we can represent interactions between agents and the environment (including other agents).

2.3.2 System Dynamics Modeling with Vensim

Vensim is an interactive software environment that not only helps System Dynamics modellers improve their abilities and productivity but also offers functionality to improve model quality [25]. Vensim (1) offers a compact, useful graphic notation, (2) supplies analysis tools to help modellers quickly construct and analyse models, and (3) provides a basic help system available locally and over the World Wide Web [26].

Compartmental models – and, in recent decades, System Dynamics models – have long played an important role in analysis of disease transmission. As an example of System Dynamic Modeling, the bird segment of the WNv model [17] is illustrated in Figure 2.2.

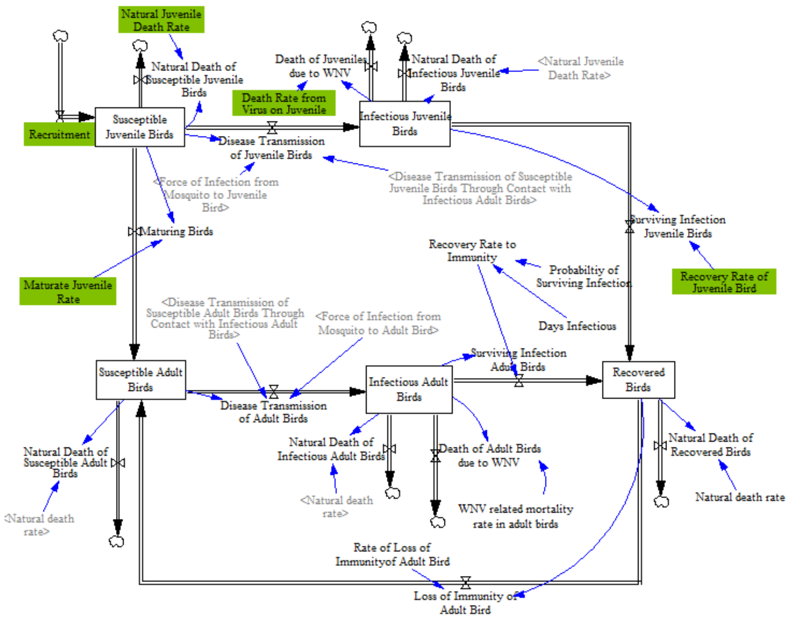

Figure 2.2: Bird segment of Yee's WNv model[17]

In Figure 2.2, the rectangles are called stocks, and represent the state of the system. Each stock accumulates the net inflow as the integral. Flows are represented by solid arrows; the value of a flow at a given point in time depends (ultimately) on the value of the stocks at that point in time, and various constants [27]. The relationship between stocks and flows can be represented by equation (1) [27].

$$\frac{\mathrm{d}stock(t)}{\mathrm{dt}} = Inflows(t) - Outflows(t) \tag{1}$$

Compartmental models are widely used in research of mosquito-borne diseases. Some models focus on human travel between different geographic regions [28]. In 2007, Brailsford, Berchi, Angelis and Mecoli constructed a System Dynamics model to study the diffusion and control of Dengue disease transmitted by mosquito- *Aedes albopictus* in Italy [14]. The paper describes the source of data sets used to create the model. The model includes a simple characterization of the mosquito life circle (including the relationship of larvae, pupae and eggs) and human's interaction in order to simulate the transmission of the Dengue virus.

2.4. The hybrid modeling approach

How can Decision Tree and System Dynamic modelling be combined to get the preferred decision based on the simulation result of the model? In 2001-2002, Kaufman and Osgood contributed a hybrid model for resource planning problems. In 2005, Osgood provided the details of the approach for implementing hybrid modeling methods, and characterized the performance advantages of the technique, relative to a classic approach in which all policy exploration and evaluated is performed inside the simulation model [11].

Within this hybrid approach, all the values of the terminal nodes are calculated by the System Dynamics model built and simulated in Vensim. In general, Vensim is given the values of each branch (e.g., In Figure 2.1, the value of a decision node is either 0 or 2), the name of the variable associated with the node (the variable name for decision nodes is "InterventionSelected"), and the time interval associated with each layer. In Amy Gao and Tony Leung's prototype [16], they provided a set of methods to calculate decision and event node's rollback value. In this thesis, I improved on their model. The detailed process will be described in the following.

Obtaining the preferred decision rule requires "rolling back" the entire decision tree. By employing the rollback algorithm, we can determine the best decision rule – which gives a sense of what should be done no matter what uncertainties apply. Specifically, for each decision node, we understand which choice is most favourable (in light of model assumptions). This often implicitly has implications for when to change our decision based on what we observe. Also for other nodes, we can have a sense as to the expected outcome (assuming that we adhere to the decision rule recommended). For instance, consider the tree in Figure 2.1. Each terminal node directly obtains its rollback value from the Vensim model. For each event node, the rollback value is the dot product of the vector of probabilities with the vector of (respectively ordered) child node values. (e.g. $\mathrm{RollBackValue}_{\mathrm{Node2}} = \mathrm{RollBackValue}_{\mathrm{Node3}} \times 30\% + \mathrm{RollBackValue}_{\mathrm{Node4}} \times 70$=-300 + (-1050) = -1350). Using the same algorithm, we can get the value of node5 (here, -840). For a decision node, the value of that node is the maximum of the value of its children (that is, of the preferred outcome). Comparing the value of each event node under the decision node 1, the biggest value is -840 so in this example, Larvaciding is the preferred decision in Week 3.

Chapter 3

Application for construction, visualization and analysis of Hybrid Decision Trees

3.1. Introduction

This chapter gives detailed information about how to implement the hybrid Decision Trees, previous prototyping done in this area, and the improvements I have introduced. The chapter also discusses the application in construction, visualization, and analysis of hybrid decision trees. It further provides discussion of previous work.

3.1.1 Previous work

The implementation described in this chapter is based upon two previous prototype systems. M.Sc. student Amy Gao built the first of these for CMPT 880 [29]. It implemented the essentials of the hybrid design, but did not support loading or saving trees, or performing a complete analysis. In her previous work, Gao introduced the idea of creating a Graphical User Interface to edit tree nodes and modify the tree structure. She used a System Dynamics model characterizing dynamics of a rabbit population as the proof-of-concept model.

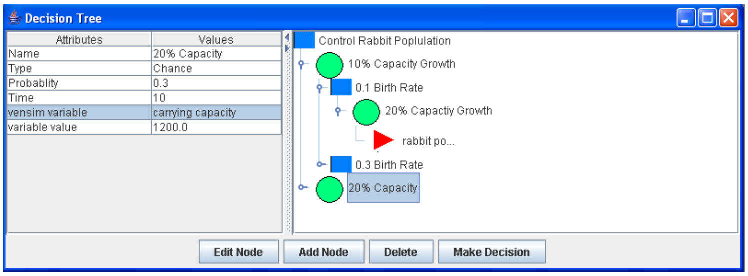

Figure 3.1: Interface created by Amy Gao

We need a friendlier interface to construct the decision structure automatically. The newer software could help users recursively create either a big tree or a small tree. Our software also provides a clearer tree view, incorporating all the information on the right hand side in Figure 3.1.
The second system was Tony Leung's command-line based prototype Decision Tree system constructed in the Java programming language for CMPT 405 [16]. Within this system, a decision tree file (XML structure) and a System Dynamics model (Vensim format) can be loaded into the system. In Leung's prototype, the system can only load a System Dynamics model from a fixed folder location (such as X: /XX/...) which the user cannot change. If the users do not have the files in the specific folder then an error message is printed. Running the prototype creates two files in the default folder. The first is an XML format file representing the parsed decision tree (but omits some key information

resulting from analysis). The other file visually depicts the tree using the Dot visual display language [30]. The Decision Tree prototype can accept the kind of tree structure depicted in Figure 2 by an XML file, with the variable names associated with the tree nodes drawn from the list of variables in the related Vensim model. All the control commands were typed from a console. There were no instructions to show what should be typed in, and no documentation maintained other than the code.
A limitation of particular significance to their project was the fact that the decision tree input file (XML structure) required manual creation. This was an urgent concern because modeling projects (and particularly projects involving zoonoses such as West Nile Virus) are typically highly interdisciplinary, and involve decisions and many stakeholders who lack a Computer Science or informatics background. As a result of these factors, the prototype required extension. The most significant priorities were improving completeness, robustness and usability.
Bai and Yee [17] constructed an aggregate model of WNv transmission that represented at a high level the stages of mosquito-borne disease among mosquitoes, birds, and humans. The model contained the following segments:

• Mosquito lifecycle (includes temperature effects).

This section of the model separated mosquitoes based on stage of life (eggs, pupae, larva, and adult), natural history of infection (susceptible, exposed, and infected).

• Bird lifecycle

The bird segment represented the lifecycle of birds that become infected with WNv. This includes progression among juvenile and adult stages of maturation, as well as the natural history of infection (susceptible, infected, recovered, and dead)

• Transmission between mosquitoes and birds

This portion of the model supports processes by which mosquitoes pass WNv infection to birds, and whereby birds infected with WNv by mosquitoes can pass the virus to mosquitoes and to other birds.

• Human infection and disease progression

The human segment of Bai and Yee's model includes natural history of infection as well as additional categories (vaccinated, exposed, asymptomatically infected, non-hospitalised patients, hospitalised patients) that involve details of WNv disease progression.

3.2. Implementation of the hybrid Decision Tree system

The system built for this project extended the prototype described in section 4.1.1, and allowed users to either load an existing decision tree (XML structure) or create a new decision tree. After running the tree through the Vensim model, we can evaluate every outcome recognized by the tree and identify the best decision at each decision point in the tree. In the output file, I added some attributes of the parsed tree structure that were originally missing but are essential for policy interpretation. The project described in this thesis further added substantial capabilities for interactive visualization and exploration of decision trees.

3.3. Structure of Decision Tree

While the system described here can be applied to an arbitrarily large set of problems, for the purpose of explaining the concepts underlying and operation of the system described here, I will use a smaller, stylized West Nile Virus disease model as an example.
In first stage of my project, I needed to design a decision tree structure that could be accepted by the Graphical User Interface. I developed an interface that can import a particular tree structure. Details will be discussed later.

3.3.1 Decision rules

A decision rule specifies the decisions that will be made in response to any sequence of events. Based on Figure 2.1, Table 3.1 lists the set of possible decision rules for the problem.

Table 3.1: Decision rules for the example shown in Figure 2.1

Rule Id	**Decision rule(The rules to deal with mosquitoes)**		**Terminal Nodes**
	Initial Decision	**Decision at Week 3**	
1	Do Nothing	Do Nothing	1,2,5,6
2	Do Nothing	If Current temperature <25 then Do Nothing else Larvacide	1,2,7,8
3	Do Nothing	If Current temperature ≥25 then Do Nothing else Larvacide	3,4,5,6
4	Do Nothing	Larvacide	3,4,7,8
5	Larvaciding	Do Nothing	9,10,13,14
6	Larvaciding	If Current temperature <25 then Do Nothing else Larvacide	9,10,15,16
7	Larvaciding	If Current temperature ≥25 then Do Nothing else Larvacide	11,12,13,14
8	Larvaciding	Larvacide	11,12,15,16

The enumeration of possible decision rules of a simple decision tree created by the Graphical User Interface for West Nile Virus has been listed in Table 3.1. In this case, users must choose between 8 decision rules. Actually, for each decision time, a decision (for simplicity, limited here to "Do nothing" and "Larvaciding") must be chosen. The table further lists a set of terminal nodes that can potentially be reached if the rule is applied; it bears emphasis that the same terminal node is typically included as part of many decision rules.

3.3.2 Loading the System Dynamics model

The System Dynamics model is presented in Figure 3.3 (in Section 3.4); the variables used in communicating to and from the decision tree are shown in a box with an orange boundary. In the dialog box in Figure 3.5, the decision nodes are associated with the variable name of "InterventionSelected". That variable relates to measures to reduce the transmission between mosquitoes and humans. It determines the method to be used to control the number of mosquitoes, such as "Do nothing", or "Use larvacide”. The event nodes are associated with the variable name of “CurrentTemperatureInCentigrade”. This represents the mean temperature observed over some period of time (e.g. 20°C), and impacts the mosquito’s lifecycle and virus maturation. The terminal nodes are associated with the variable named “NegativeOfCumulativeWNVSymptomaticCases”, which is just the negative of the cumulative count of people presenting with WNv symptoms in the model.

In order to get the smallest rollback value of the stock called “CumulativeWNVSymptomaticCases” (which is always non-negative), the rollback algorithm attempts to maximize the value of the negative of “CumulativeWNVSymptomaticCases” (i.e., try to bring this non-positive value as close to zero as possible) during rollback.

Figure 3.2: Load System Dynamics Model interface

The GUI implemented in this thesis allows the user to load the Vensim model from any location in their computer. Afterward, the users can either use an existing tree (specified in XML format) or create a new decision tree by using the friendly interface.

3.4. Implementing the hybrid method

To implement the hybrid method described earlier, basic logic is required between the interface and the Vensim model the user loaded. This component uses the tree structure specified by the user, identifies the sequence of decisions and events required for decision scenarios, and calls the Vensim model to run the simulation for the scenarios, and then calculates the roll-back value for each terminal node. It has several stages:

1. Load the appropriate Vensim files
2. Use Vensim API and Java code to
   a) Save the values of the parameters in Vensim
   b) Simulate each decision scenario from root to terminal node in Vensim
   c) Sets roll-back value for each terminal node
   d) Perform roll-back

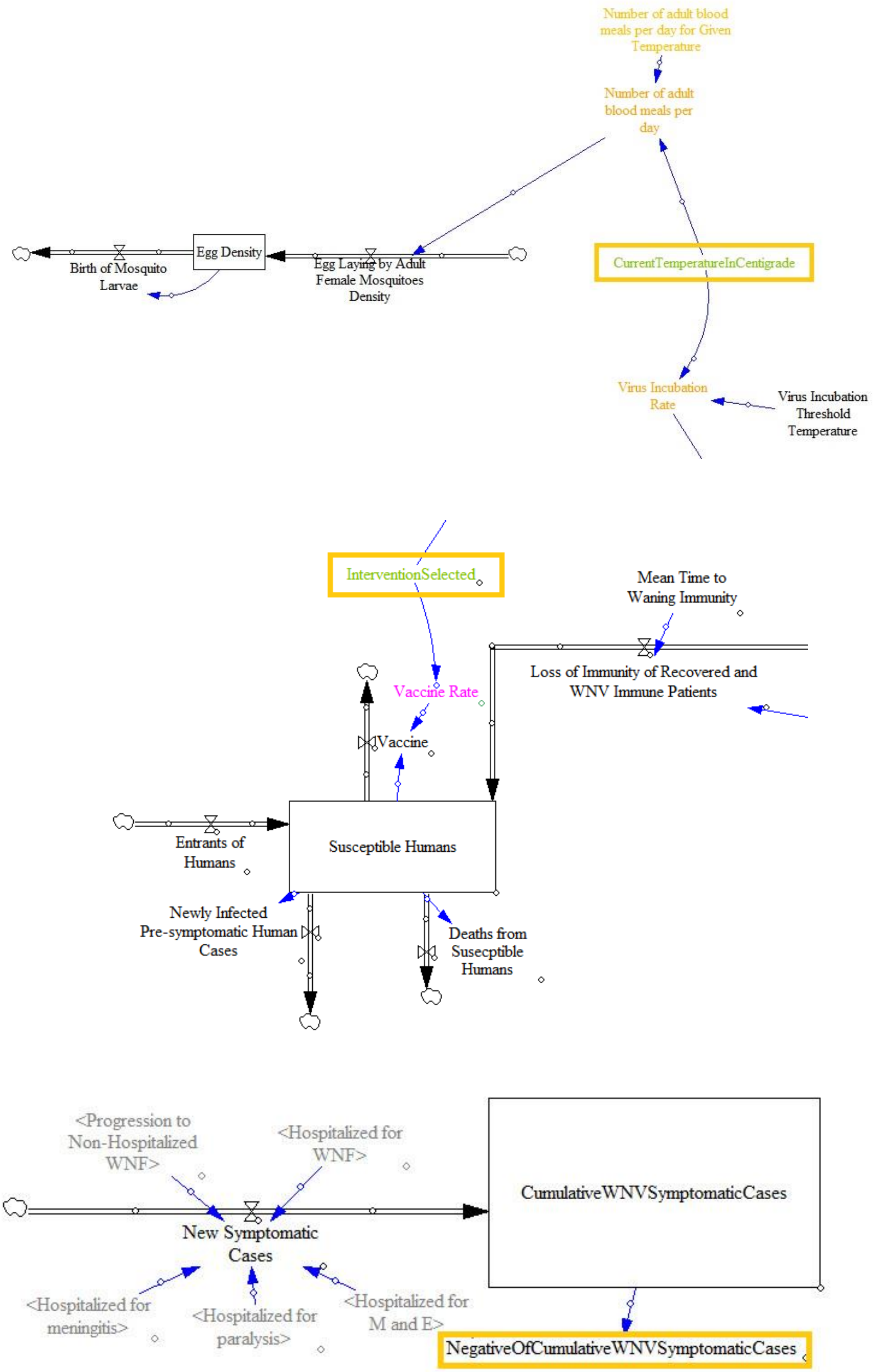

Figure 3.3: Three Fragments from the West Nile virus System Dynamics Model

3.4.1 Preparing the Vensim model for simulation

In a decision tree, for a specified decision rule there will be a set of terminal nodes. Each terminal node is associated with a specific scenario that proceeds from the root node to the terminal node, and which includes a unique sequence of decisions and events over time. For instance, consider rule 1 in Table 3.1. The possible scenarios associated with rule 1 are associated with the terminal nodes 1, 2, 5, and 6. Other decision rules will be associated with different unique – but overlapping – sets of terminal nodes. To determine the best such decision rule, it is necessary to calculate the roll-back value for all nodes of the tree. The first stage of this is to determine the roll-back value for each terminal node.
Vensim requires a dynamically linked library – called "Vensim32.dll"– which provides some functions to programmatically control – and access variable in – the Vensim model. In that file, in order to set a variable value in Vensim by using external language, we can use Vensim *command* such as "SIMULATE>SETVAL" to lead Vensim set parameters' value. To allow variable values to be changed at different time periods in a given simulation, the particular variables to be changed must be marked as "Gaming" variables in the Vensim model, and the system enables the Vensim GAME mode[1].
To set a variable in the Vensim model to be a gaming variable, the user can click the "Equations" button from the tool bar and then click the variable which needs to be changed. They can then select "Auxiliary" and "Gaming" in the Type drop-down list (for more details, please see [31]).

In Figure 3.5, the variable "InterventionSelected"[2] and "CurrentTemperatureInCentigrade"[3] are all of GAME type, so that they can be set from the hybrid decision tree software described here.

[1] Vensim GAME mode is a mode of simulation in Vensim during which the user – or a program – interacts with the model to make decisions as the simulation progresses [31].

[2] "InterventionSelected" in the model includes several approaches to control amount mosquitos.

[3] "CurrentTemperatureInCentigrade" indicate different levels separated by different centigrade temperature.

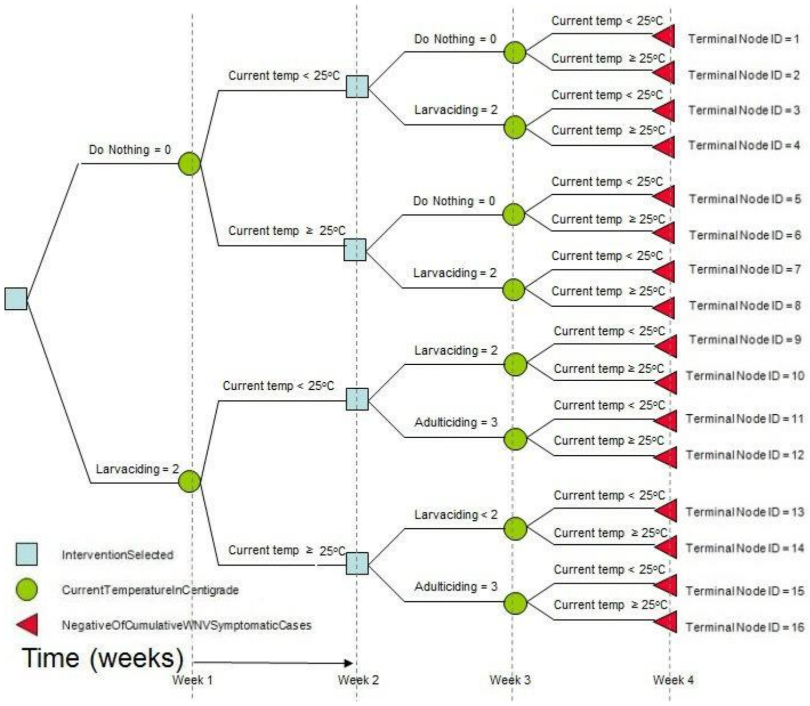

Figure 3.4: Full decision rules

3.4.2 Construct Decision Tree by using interface

It is very labor-intensive to create an XML file to specify a decision tree, especially if the depth of the tree is very large, or if it needs to be checked for consistency between multiple nodes. To ease the specification of decision trees, I designed and created a dialog box to help users create the decision tree. This dialog box is presented in Figure 3.5. In this dialog, the user can provide information regarding the decision tree, such as the name of the tree.

In order to be specified concisely, the tree which is to be created by this GUI must exhibit a high degree of regularity; the simple implementation here assumes considerable uniformity. Most notably, the interface requires branch count and names, associated Vensim variables, and branch to value mappings of a particular type of node (e.g., decision node or event node) to be identical at all levels. Similarly, all terminal nodes must share a common associated Vensim variable. The tree should alternate between Event and Decision layers, allowing the users to choose the type of the root node to be either a decision node or an event node. The user may (separately) select the variable names to be associated with decision, event and terminal nodes in drop down lists, which can be chosen by the users. This ensures that the users will not specify a variable name which is not defined the Vensim model.

The dialog box provides users the ability to define the tree depth. The tree specification further requires a “start time” value which can also be set within this Dialog box and is used to specify the initial time for the tree (which may be different from that of the Vensim model). An error check function is implemented within the graphical user interface which discovers some typos and identifies and prevents problematic user operations.

Each event branch and decision branch to be replicated throughout the tree can be specified by the users. By using the “Add” and “Delete” button, it is easy for users to modify the branch specification.

When the users input the information associated with event branches, that information must include the probability of each branch, and the system can help users to automatically fill the probability. To allow some flexibility in tree structure over time, the dialog box does provides the user with the option of specifying different probabilities to associate with a given event node branch at different levels of the tree. After the “Run Tree” button has been clicked, a decision tree will be constructed.

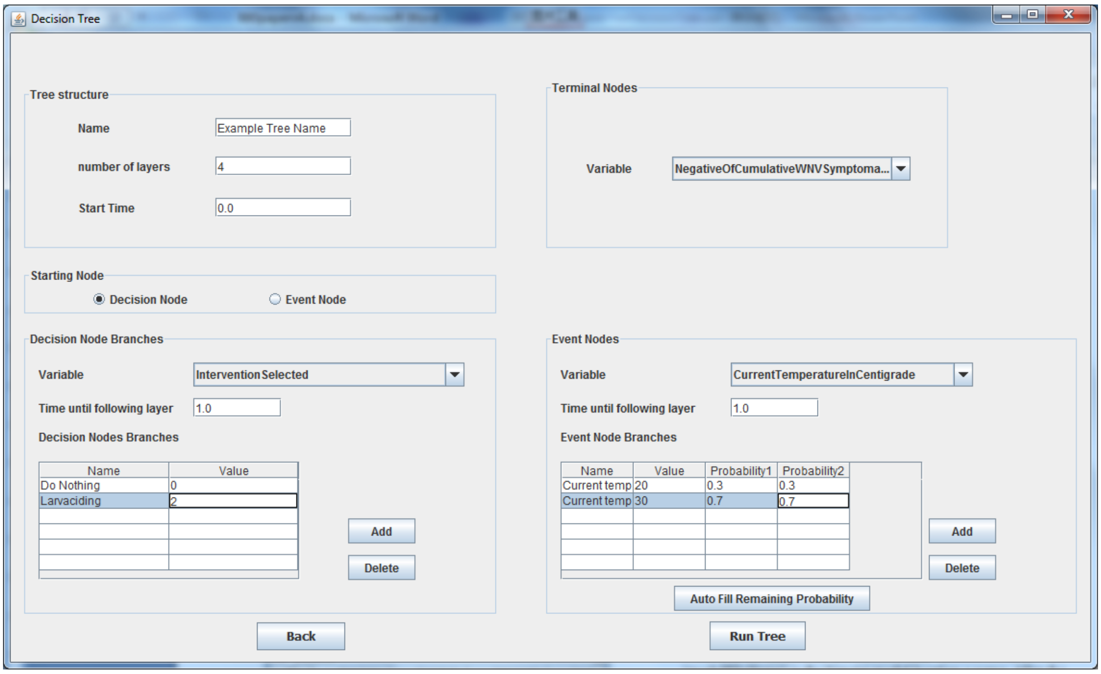

Figure 3.5: Create Decision Tree interface

3.4.3 Process of interaction with the software

To specify a decision tree, the system provides two options. The first one is using an existing decision tree, which is in XML format. The structure of the tree does not have to be uniform or symmetric. The second way to build a decision tree is to create a new tree, through the interface described below.

```
createSymmetricTree (type_Of_Node, int depth, int iCurrentLayer, double currentSimulationTime)
{
     if    type_Of_Node is TerminalNode
            return new TerminalNode(terminalNodeTemplate.strSimulationVariableName(), currentSimulationTime)
    else if    type_Of_Node is EventNode
    {
            return new EventNode(eventNodeTemplate.strSimulationVariableName(), currentSimulationTime,
                  [ For i=1 to eventNodeTemplate.countEvents();
                        EventEdge(eventNodeTemplate.eventValue(i), eventNodeTemplate. particularEventProbabilityForLayer
(i,iLayer), createSymmetricTree (notEventLayer, depth,iCurrentLayer+1, currentSimulationTime+
eventNodeTemplate.timeUntilFollowingLayer()))    ])
    }
    else{
          return      new DecisionNode(decisionNodeTemplate.strSimulationVariableName(), currentSimulationTime,
                [ For    i=1 to decisionNodeTemplate.countParticularChoices();
                         DecisionEdge(decisionNodeTemplate.particularChoiceValue(i), createSymmetricTree (notEventLayer,
depth,iCurrentLayer+1, currentSimulationTime + decisionNodeTemplate.timeUntilFollowingLayer()))    ])
        }
}
```

Figure 3.6: Pseudo code for creating symmetrical tree algorithm

As in Figure 3.5, after specifying the tree structure, there is a button called “Run Tree”, which runs the algorithm specified in Section 3.4.

We illustrate this process using an example regarding West Nile virus in Figure 3.5, where the purpose is to find of whether or not to put in place an intervention when the temperature is closer to 20°C or 30°C, based on the outcome specified (the negative of `cumulativeWNVSymptomaticCases`[4]). The decision is either to do nothing or to larvacide. The uncertainties are current temperatures that are either specified as being closer to 20°C or 30°C. The unit of the time is weeks, so each layer’s time interval is set equal to 1. As indicated in Figure 3.8, this tree has four layers. From the root node to the terminal node, each scenario will traverse 2 event nodes and 2 decision nodes. For the first event, we consider the probability of hot weather (closer to 30°C) as being 0.3; for the second (later) event, as 0.7.

For the sake of the example, I will choose to create a new tree using the interface. The information regarding the tree is represented in Figure 3.5.

On the basis of user specified information regarding tree nodes and branches, the business logic creates a tree structure using the algorithm in Figure 3.6. The notation [*For i=n to m*; *expression (i)*] denotes an array of values whose elements consist of values of the expression evaluated on each value i, $n \le i \le m$.

Before this step, the Vensim API was loaded and Vensim was issued the command of "SPECIAL>LOADMODEL|"+`modelFileName`. The “modelFileName” is the Vensim model location + name. After the system loads the System Dynamics model, and because Vensim API supplies the

[4] Cumulate all the infected people which has WNv symptomatic.

get_varnames[] function, the variable names are loaded into the model, and presented in the drop down list of Figure 3.5.

In the decision tree software, to get the consequence associated with a specific scenario proceeding from the root node to a terminal node, the handler uses recursion to run a simulation for the entire decision tree.

```
evaluateTree (TreeNode currentNode, SimulationActionStack simulationActionStack)
{

     if (currentNode==terminalNode)
          rollbackValue = run(currentNode .strVariableName, time)
     else if (isEventLayerIfNonLeaf)
          rollbackValue=dotProduct(
               [ For   i=1 to currentNode.countEdges();
                    evaluateEdge(currentNode.getEdge(i), time) ],
                    currentNode.getEdges())
    else
          rollbackValue=max(
               [ For   i=1 to currentNode.countEdges();
                     evaluateEdge(currentNode.getEdge(i), time)])
     currentNode.setRollbackValue(rollbackValue)
     return rollbackValue
  }

evaluateEdge(TreeEdge currentEdge, double time)
{
     simulationActionStack.push(new function(sim){sim.setValue(currentEdge.getSimVariable(),currentNode)},
               time)
     double value = evaluateTree(currentEdge.getDestinationNode(),simulationActionStack)
     simulationActionStack.pop()
     return value
}
```

Figure 3.7: Pseudocode for simulation decision tree algorithm

This algorithm is actually a stack framework to contain a decision scenario. As above, the notation [*For i=n to m; expression*(*i*)] denotes an array of values whose elements consist of values of the expression evaluated on each value i, $n \le i \le m$.

In the original prototype, the id number of each node and edge were drawn from the same number sequence. This decision caused the algorithm to create a tree structure that was difficult to interpret. I separated the calculated tree node id and tree edge id.

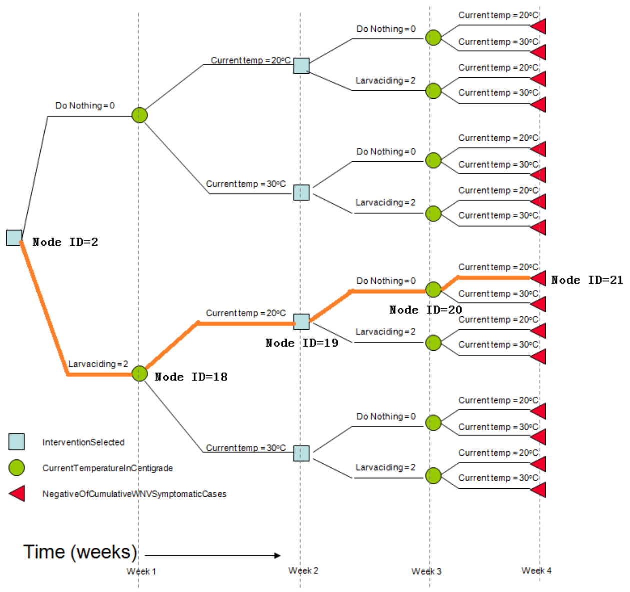

Figure 3.8: Representation of a specific decision rule

Table 3.2: Partial outcome of the chosen path in Figure 3.8

The branch value	Node Information (XML Format)
	<TreeNode type="DecisionNode"> <treeNodeID>2</treeNodeID> <time>0.0</time> <variableName>InterventionSelected</variableName> <rollBackValue>-0.006905681863427162</rollBackValue> <nameOfBestDecision>Do Nothing</nameOfBestDecision> <idOfBestDecision>10000</idOfBestDecision> <selected>true</selected>
Larvaciding = 2	<Edge type="BasicEdge"> <edgeID>10015</edgeID> <edgeName>Larvaciding</edgeName> <value>2.0</value> <selected>false</selected> <sourceNodeID>2</sourceNodeID> <destinationNodeID>18</destinationNodeID> </Edge>
	<TreeNode type="ChanceNode"> <treeNodeID>18</treeNodeID> <time>1.0</time> <variableName>CurrentTemperatureInCentigrade</variableName> <rollBackValue>-0.00693912477698177</rollBackValue>

	<selected>false</selected> <sourceEdgeID>10015</sourceEdgeID></TreeNode>
Current temperature=20	<Edge type="ChanceEdge"> <edgeID>10016</edgeID> <edgeName>Current temp</edgeName> <value>20.0</value> <probability>0.3</probability> <selected>false</selected> <sourceNodeID>18</sourceNodeID> <destinationNodeID>19</destinationNodeID> </Edge>
	<TreeNode type="TerminalNode"> <treeNodeID>21</treeNodeID> <time>4.0</time> <variableName>NegativeOfCumulativeWNVSymptomaticCases</variableName> <terminalValue>-0.0038220123387873173</terminalValue> <selected>false</selected> <sourceEdgeID>10018</sourceEdgeID> </TreeNode>

Table 3.2 represents a part of the outcome file. Because node 19 and node 20 are similar to node 2 and node 18 (although associated with different rollback values), the description is not repeated here.

Table 3.3: Tree node attributes

Attributes of nodes	
TreeNode type	The type of tree node (DecisionNode/ChanceNode/TerminalNode)
treeNodeID	The id number of the treenode. 1 is default for tree id., so it begins from 2
time	The time from the root node to the current layer
variableName	This node specifies the Vensim variable that is associated with this node
nameOfBestDecision **(**Decision nodes only)	The decision name of best branch
idOfBestDecision **(**Decision nodes only)	For decision nodes, the branch id of the best decision
Attributes of edges	
Edge type	The type of tree edge (DecisionEdge/ChanceEdge)

edgeID	The id number of the tree edge.
edgeName	The name of the decision or event
value	The value of the edge
Probability **(Event nodes only)**	The probability of this outcome
sourceNodeID	The node id of this branch's source
destinationNodeID	The node id of this branch's destination.

3.4.4 Identifying a preferred decision rule

Policy makers want to find an optimal decision rule. The previous section described a means by which each possible scenario that could occur is evaluated. After the Vensim model simulation and based on each scenario's value of variables, the decision tree software computes all of the terminal node values. We then need to figure out how to get to the preferred decision rule by using this hybrid method. The rollback algorithm is listed in Table 3.4 and is based on the principle of backwards induction [32]. For the ease of presentation, the name Rollback Value will be denoted as RB.

Table 3.4: Formula to calculate roll back value for each node

RB (Terminal Node t)=$f_{\text{System Dynamic outcome}}$ (time) // BaseCase	t in RB (t) denotes the terminal node. The roll-back value of terminal node is the outcome of the Vensim model which has run the scenario from the root node to the current terminal node.
RB (Event node e)=$\sum \mathrm{P(edge)} * \mathrm{RB}(\text{getDestinationNode(edge)})$	e in RB (e) denotes the event node. P(c) means the probability for each possible outcome of this event (i.e. for each child of this node in the decision tree). For chance (event) node, the backwards algorithm is the weighted sum of the rollback values of the children (i.e. the dot product of two vectors defined over the children: the vector of probabilities with the vector rollback values).
RB (Decision node d): $\max_{\text{children c of node d}}$(RB(c)) PreferredDecision(Decision node d) = arg $\max_{\text{children c of node d}}$ (RB(c))	PreferredDecision(is the best branch after calculating the largest roll-back value from the decision node's children.

3.5. Evaluation

When compared to the prototype version, the system created in this project exhibits improved features, especially with regards to robustness and usability. By adding graphical user interfaces, we help enhance the systems' accessibility, to allow those without strong computer background – such as policy makers and analysts – to take advantage of this approach. The accessibility and usability of the approach is also strengthened by permitting the user to build a custom decision tree without requiring familiarity with XML. By extending the error check function, we can reduce the likelihood of errors. Most importantly, after extending the attributes in the output file (XML format), it is easy to see which decision is the preferred decision.

Even though this system still exhibits limitations, it is the first interactive, end-to-end system using a hybrid approach that combines the System Dynamics and decision analysis method together for policy making. This system can be used not only in Public Health area, but in any area which uses Vensim simulation models to examine dynamically complex decision problems under uncertainty. Moreover, by using this system, one can greatly reduce the cost for performing such decision making. Right now, this system has been employed for a policy making organization in Calgary, and is intended for use with the Saskatoon Health Region and Saskatchewan Ministry of Health.

Chapter 4

Expanding the West Nile virus System Dynamic model

The previous chapter provided an introduction to the quantitative elements of Decision Trees. In this chapter, we present several requirements for the work laid out in this thesis.

We provide here a discussion of handling differences in the date formats between Vensim and Excel, a summary of the shortcomings in last version of the WNv model dealt with by our work, improvements in the internal variables of the model, and the approach taken to calibrate the model.

4.1. Time unit of the West Nile virus model

The time unit is a key concept in simulation modeling, representing the amount of real-world time associated with a unit increment in the simulation model (one day, one hour, or one month, etc.).

In our case, we sought a way of mapping between real-world dates and simulation model times. To do this, we made use of Excel's encoding of dates[5].

The aggregate data in Yee's [17] model was set from 01/01/2004 to 31/12/2009;. Within Excel's encoding, this corresponds to the time range 37987 to day 40178. The simulation result for each 1/8 day is obtained (as Table 4.1) by running the model ¸so as to accurately capture some of the more rapid dynamics involved.

Table 4.1: Parameter setting of the Yee [17] WNv model

Name	Value	Description
Initial time	37987	Model run start time
Final time	40178	Model run end time
Time step	0.125	Time interval = 1/8 day (3 hours)
Units of time	Day	Model time unit

4.2. Problems with the original WNv model

Inspection of the model by Yee [17] identified several components that were missing or incorrect. We discuss such major issues below.

In the bird segment (Figure 2.2), the recruitment (rate of arrival of juvenile birds) was treated as a fixed rate of flow, regardless of season. This poses problems because American crows lay eggs only in the spring of each year [33], and because fledgling birds – who are unable to fly – are particularly susceptible to mosquito predation, and may serve as an important element in the WNv transmission pathway.

[5] To illustrate this mapping, the reader can open an empty Excel file and input 1 in A1 cell. Now we need to transfer number 1 in cell “A1” into a date. We click “A2” and type in this formulat “=DATE(YEAR(A1),MONTH(A1),DAY(A1))”. In “A2” it represent “1900-01-01”,

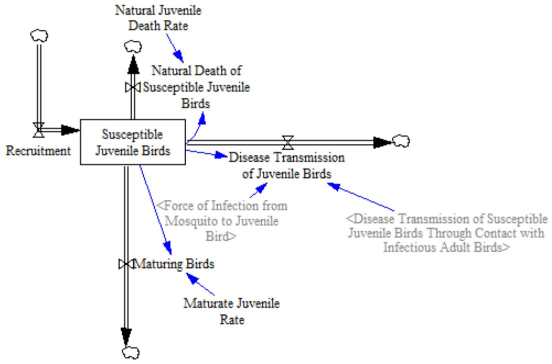

Figure 4.1: The susceptible bird part in Karen's model [17]

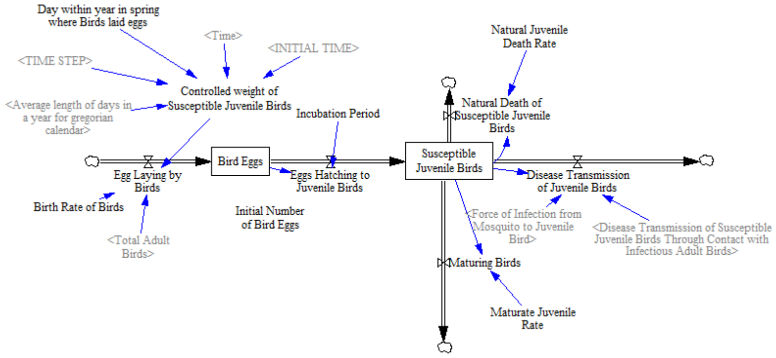

Figure 4.2: The susceptible bird part in the updated model

A controlled recruitment for "Susceptible Juvenile Birds" is an important element of the WNv model [17]. American crows lay eggs only at the end of May. By using that information, we can simulate that in the spring of each year, the number of "Susceptible Juvenile Birds" will be adjusted regularly. Figure 4.3 depicts the square-wave used to regulate egg-laying [33] in the extended model.

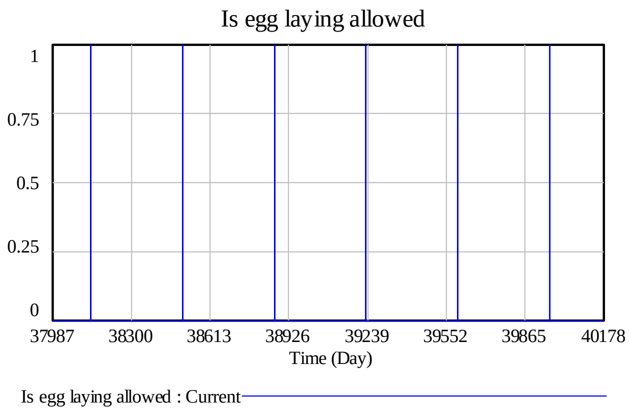

Figure 4.3: Addition of "BirdEggs" controls the number of "Susceptible Juvenile Birds."

4.3. Model extension and refinement

4.2.1 Refinement

All the variables marked with colored frame in Figure 3.3 were listed, and Dr. Nate Osgood was consulted to determine which variables needed to be filled. Based on Yee's WNv model [17], we split the variables into four types (Table 4.2).

Table 4.2: Variable segregation from Figure 3.3

Variable type	Named example	Source of the value
Not signed	Average length of days in a year for Gregorian calendar	The value is easily calculated.
OK	Probability of surviving infection (birds)	Literature
NOT OK	Juvenile (birds) death rate from virus	[32]
Calibration needed	Source Reduction Death Rate of Immature Female Mosquitoes	To be determined by calibration

Identifying values for the "NOT OK" variables was the main target of our optimisation approach. Values were obtained from three sources:

1. Experts (email or meeting) (Nathaniel Osgood, Dr. Tasha Epp. Associate Professor, Phil Curry Zoonotics Diseases Consultant and Provincial West Nile Virus Co-ordinator)
2. Literature search
3. Data from Saskatoon Health Region or other data resources. Because of the focus of the model, data from Saskatoon, Saskatchewan was our main interest (Table 4.3).

Table 4.3: Model-Infected WNv populations in Saskatoon, SK

	WNv Vector	Type
Reservoir Species	Bird	Corvids (American crow, magpie[6])
Vector	Mosquito	*Culex tarsalis*
Dead-End Hosts	Human	Saskatoon residents

In 2004, the first major North American epizootic of WNv in California, Reisen et al. [34] realized that WNv epizootics played remarkable role in affecting American crows, crow roosts and crow mortality [35]. American crows produce highly elevated viremias from the time shortly after infection to death. In Saskatoon, *Culex tarsalis* is the species of mosquito that transmits the West Nile virus to people [36]. While research is ongoing into other reservoir species, both avian (e.g., American Robin) and reptilian (e.g., frogs), it is widely believed that *Culex tarsalis* mosquitoes predominantly become

[6] Some researchers (Kilpatrick et al. "Host heterogeneity dominates West Nile Virus transmission") believe that other bird species may play a role.

infected by biting infected American crows; mosquitoes then bite people and transfer the disease to humans (who remain dead-end hosts; that is, they do not transmit the infection on to other mosquitoes or other animals).

4.4. Filling in the WNv model's omissions

In the mosquito segment of the WNv model [17], it was found that “Exposed Adult Female Mosquitoes” and “Endogenously Calculated Infectious Mosquitoes” stocks aggregated into an unrealistically large number throughout the year because the diapause (“hibernation”) behaviour in these two stocks was not considered. Figure 4.4 represents the hibernation flow I added to the mosquito segment of the WNv model [17].

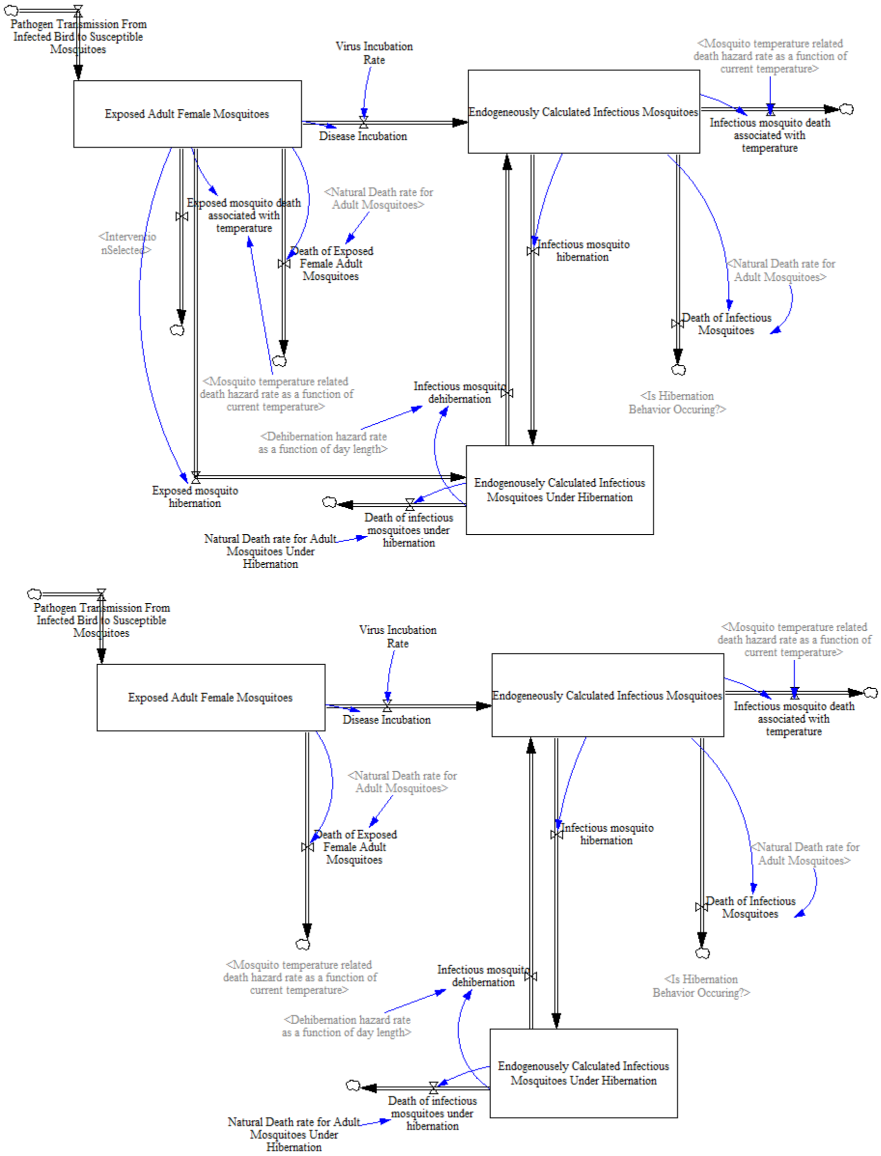

Figure 4.4: Result of adding hibernation flow to the mosquito segment of the WNv model[17]

4.5. Extending the calibration

4.5.1 Calibration machinery in Yee's model

In order to find the best match compared to the data simulated from our model and historic numbers. I adjusted the calibration with a discrepancy function given by the following formula (2) [37]:

Table 4.4: Description of the parameters in equation (2)

Parameter	Description	Example
w	"A dimensionless weight that can help quantify the relative degree of importance we place on this particular discrepancy" [37]	Controlled weight for discrepancy in human cases confirmed over the past year
m	The simulation model value corresponding to the historic value	Number of human cases confirmed over exactly the past year
h	Historical value	Number of human cases confirmed historically

$$w\left(\frac{h-m}{average(h,m)}\right)^2 = w\left(\frac{h-m}{\left(\frac{h+m}{2}\right)}\right)^2 \qquad (2)$$

Furthermore, changes were made to account for the fact that the data on the "historical number of human confirmed cases" is given by year (rather than day). Although the framework supports use of differing weights, the model is currently using the uniform rates for the weights.

4.6. Adapting the System Dynamics model into an agent-based model

The Vensim model discussed above is a System Dynamics model; it focuses on aggregate numbers over time in a specific area. In Saskatchewan, West Nile virus rates could vary across regions depending on local environmental conditions (such as eco-zones and changes in weather conditions and presence of mosquito habitat). The mobility of mosquitoes and birds in one area can allow West Nile virus in one region it to transfer to it to a neighboring region. To simulate the infection across multiple areas – and flow of mosquitos and birds between such areas – we employed AnyLogic software. We created a patch as an agent and imported the WNv model [17] as the agent dynamics, allowing the disease to be represented in neighbouring patches (Figure 17). Additional mechanisms were added to support transfer of disease (via mobility in vectors and reservoir species) between multiple areas, as well as visualization.

4.6.1 2D spatial embedding

For the embedding under consideration, we need to simulate WNv transmission between each patch and its neighbours. To accomplish this, we employed 2D discrete environment space type, which partitions space into a rectangular array of cells.

4.6.2 Implementation

In the model, three components need to be monitored especially closely: prevalence of WNv among birds, prevalence among mosquitoes, and cumulative number of infected individuals. Figure 4.5 shows different depth of color in each grid. By searching the color scale on the right hand side, users can easily get the estimated number in the grid. The grids at the upper left represent the magnitude of each component. The simulation allows the user to select which of these elements they wish to display at any one time; radio buttons allow users select the variables they want to monitor; the charts at the bottom right of Figure 4.5 aggregates across the model.

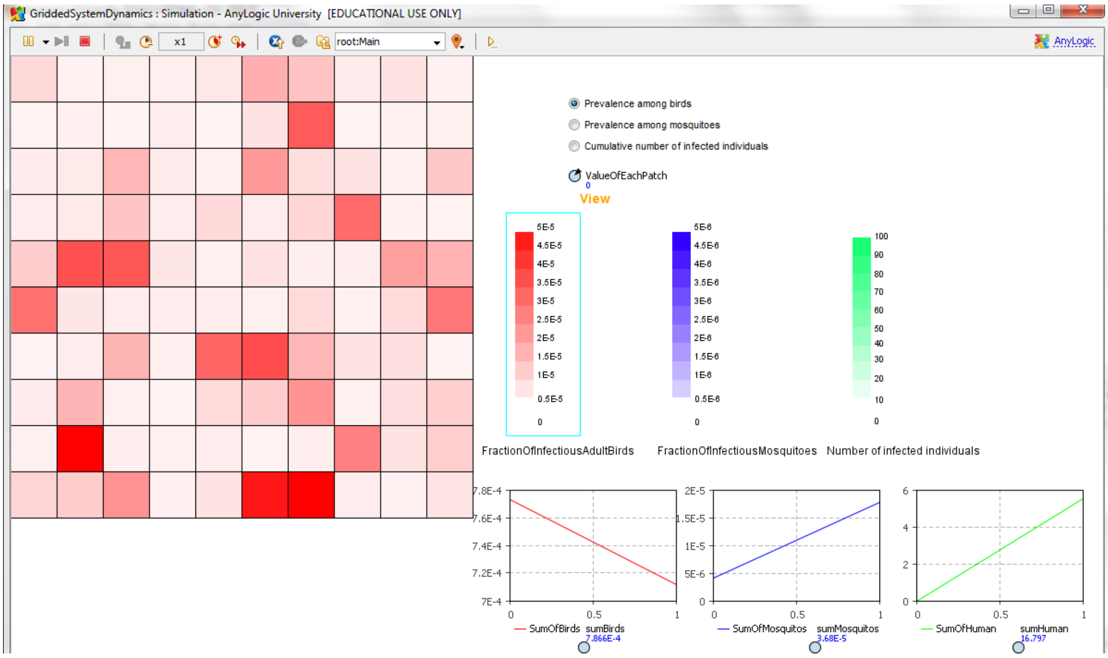

Figure 4.5: Interface of the gridded WNv agent based model

To find the value of a patch, the user clicks on the patch and the value itself is presented in the “Value of Each Patch” at the bottom of the radio buttons.

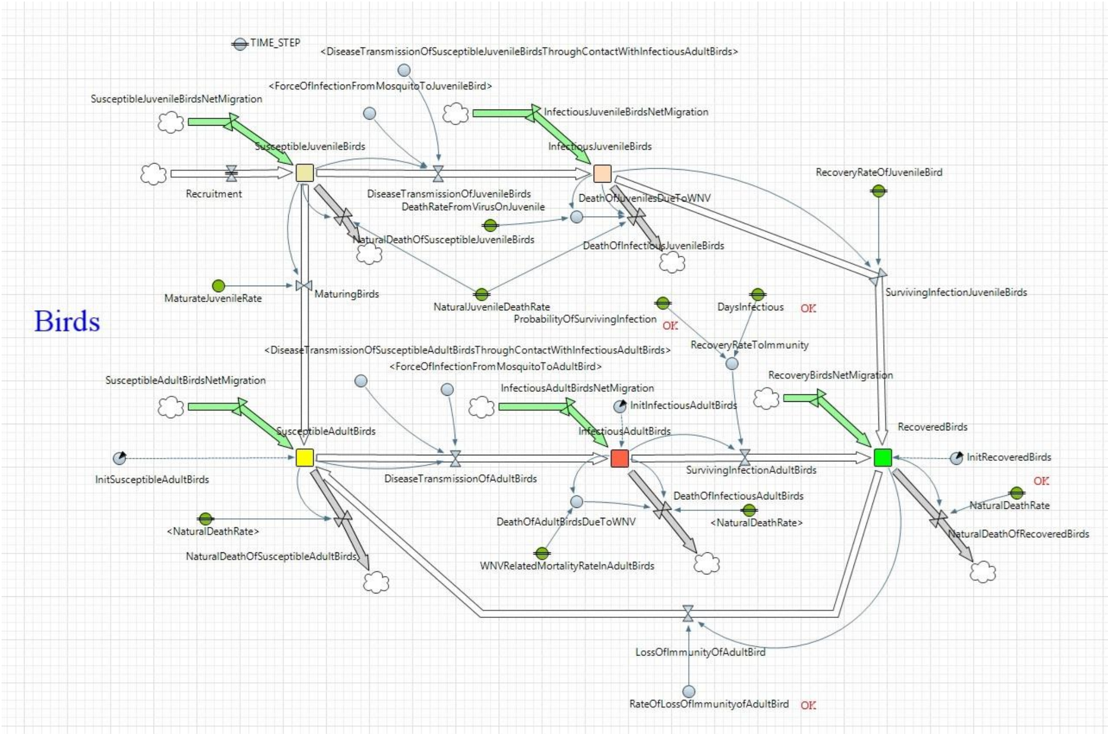

Figure 4.6: The System Dynamics model in each patch

Each patch (Figure 4.6), includes a distinct – but cross-coupled – System Dynamics model of WNv. Figure 18 only depicts the segment for birds. Each stock could be linked up with a parameter. In the Main view, we can set an aggregator to calculate the summary of a specific stock across all patches.

4.7. Experimental results

In Yee's WNv model [4], the fact that human cases were only recorded yearly impaired calibration. We now have daily data, which were not available while Yee's model was constructed. I used this accurate daily data set instead of the yearly data set, and the accuracy of the calibration was significantly improved.

The accuracy of the model was also significantly improved by looking for values of variables and modifying the model construction. The mosquito sector of Yee's WNv model [4] included hibernation flows. But this part was not completed, there were still some stocks missing in Yee's model. After adding hibernation data flows to lead mosquitoes to other stages, all of the important stocks in this model decreased to a reasonable range.

To minimise the total discrepancy, we set the payoff definition of the model. The closer a payoff is to 0, the better the result. The difference in the payoff indicates that filling values and employing the other modifications described enhanced the accuracy of the model. There are some limitations, such as some parameters' value are not very accurate. We also need to find better values for some variables from literature. While Figure 4.7 indicates that the changes significantly improve the model match to empirical data, it should be noted that because the metric used involves the square of the discrepancy, the degree of imprecision rises with the square root of the negative payoff (rather than linearly with the payoff).

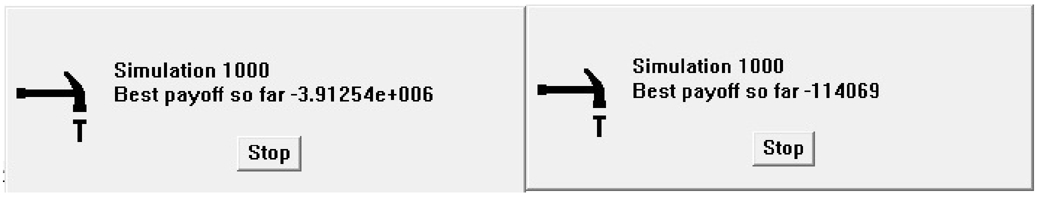

Figure 4.7: Payoff before and after WNv model [1] modification.

4.8. Discussion and conclusion

I improved the accuracy, robustness, and usability of the WNv model [1]. By inserting variables based on literature, and by adding new structure to capture important processes such as bird egg laying and hibernation, I enhanced model accuracy. By extending the model, I was using AnyLogic to create a hybrid model. The model is an agent based model, each agent consists a Systems Dynamics model. The usability of the model was strengthened by permitting users to indicate the patch numbers and variables they want to monitor.

Chapter 5

Design and implementation of Decision Tree visualization

5.1. Background

Right now, the software to implement the hybrid modeling approach produces XML format output. This XML file includes the tree structure and detailed information for each branch. However, an XML file is difficult to read for anyone, but especially for those lacking computational training. It is also hard to update. Leung's [16] prototype version of the software can generate an output file (Dot format-a plain text graph description language [38]) which supports a very simple tree structure by visualization, but this file is very messy and visually extremely confusing for even moderate-sized trees. For those reasons, I tried to implement an interactive graphical visualization tool for the decision tree structure by using an open source information visualization toolkit called "Processing" [39] which is a programming language to easily create images based on math formulas and the Java language. But investigation with this system revealed pronounced shortcomings when running large tree structures: Transferring a large tree structure into an image takes an enormous time. Fortunately, as pointed out by another student, there is a better visualization tool, namely "Profuse", that we used for our work. This software framework can create images, animations and interactions. It also supports creating interactive data visualizations. Also, tools for website construction and interaction, including Action Script and the Adobe Flash Player [40] are also provided.

5.2. Motivation

While other mechanisms described in this thesis are responsible for populating tree information, the challenge addressed in this chapter is how to present this information in an easily understandable and usable fashion. Specific requirements were required to address common needs:

(1) The system should provide a clear and friendly tree view to represent the tree structure, such as those created by the software described in Chapter 4.
(2) The system should provide either an entire tree view or the detail for each leaf or branch.
(3) Given that decision trees can grow very large with even just a modest depth, the system should be able to gracefully handle trees composed of 10,000s (and, optionally, more) of nodes.
(4) Regardless of whether it originated when creating the tree or as a result of the analysis, the node-specific information (such as roll back value, each node's name or value) of a decision tree for each scenario should be easy to see.
(5) As a key requirement, the system must support highlighting the preferred decision.
(6) The system must display a timeline.
(7) For a given scenario, the user should be able to view the behavior over time of any simulation model variable.

5.3. Implementation

Based on the system requirements, I employed the API of “Prefuse” toolkit. In order to reach requirement (1), the first step is to read in XML file the format output by the Decision Tree software. The process to construct this tree for viewing is different from that used for the Decision Tree software described in Chapter 3. In the previous chapter, we used recursion to build the Decision Tree for analysis. But in this tree, the XML has already been populated with tree information. Instead, “Perfuse” reads XML (such as that shown in Table 3.2) line by line, using a process similar to that illustrated in subsequent figures.

```
<TreeNode type="DecisionNode">
…
<childrenEdgeList>
<Edge type="BasicEdge">
…
</Edge>
<Edge type="BasicEdge">
…
</Edge>
</childrenEdgeList>
<childList>
<TreeNode type="ChanceNode">
                              …
<childrenEdgeList>
<Edge type="ChanceEdge">
                                      …
</Edge>
```

Figure 5.1: The structure of XML

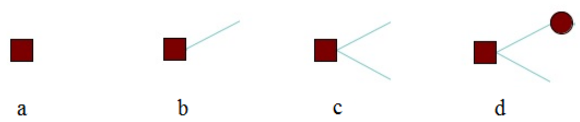

Figure 5.2: The process of Prefuse constructing the read tree

Figure 5.2 exactly presents how we construct the tree in Prefuse. When the system reaches a TreeNode, it will create a tree node; if the tree is null, then create a root node. After parsing the information included in the node element, it creates a node. Edges are read in by following Edge List, until it reaches another node in the file, at which point it will add it to the first empty edge of the parent node. For requirement (2), users could zoom in and out the tree structure view by using mouse scroll wheel to see the whole tree on details of the tree. This toolkit could afford large XML described in requirement (3) into tree structure view.

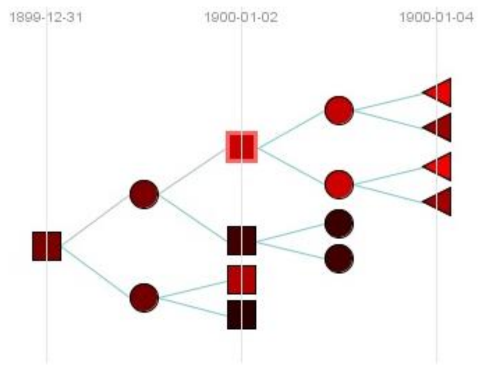

Figure 5.3: The color layer of the tree structure

The rollback value of each node is a crucial value of the decision tree for policy making. Those using the system will likely want to make the rollback value clear at a glance. In the coloring scheme used, a node which has lighter color means the rollback value of this node is closer to 0.
Chapter 3 characterized the information associated with each type of node and edge. To reach requirement (4), we added a “mouse over” event to let the information of a node or an edge be readily shown in a label at the bottom.

Table 5.1: Node information to present during mouse over events

Description	**Node Information Type**	**Node Information Name**
Node type	String	Name
Node id	Integer	Id
The time in the model when this node had been reached	Double	Time
The name of the simulation model variable associated with this node	String	variableName
Rollback value of this node	Float	rollBackValue
Terminal node’s rollback value	Float	terminalValue
If this node had been selected	Boolean	Selected
The edge id of the best decision associated with this decision node	Integer	idOfBestDecision

As requirement (5) described, if a node has been chosen, the path from root node to the current node will be highlighted and the name of the edges on that path will be indicated. Furthermore, the chosen node will be highlighted. When the tree is very large, this function allows policy analysis and makers to get a general idea of the outcomes of particular decisions in light of preceding events.

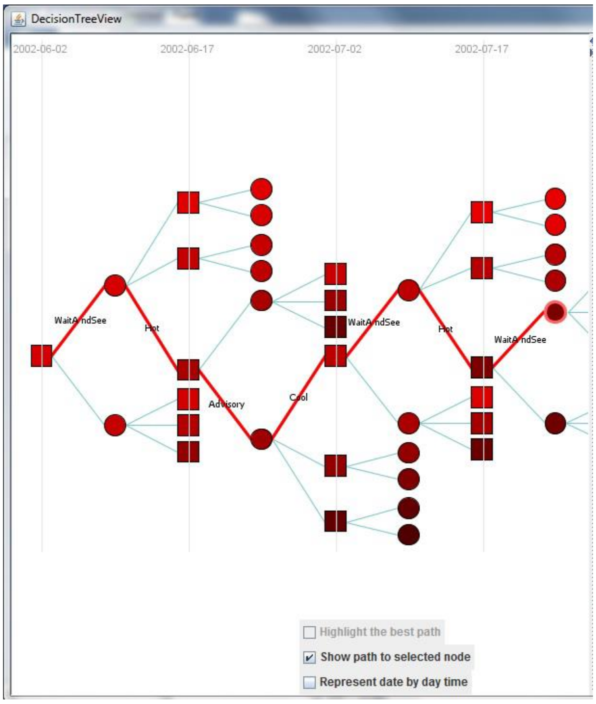

Figure 5.4: Show the current path

When decision makers want to know the preferred path which is given by the Decision Software, they can mouse over each node and take a look at the detail of “idOfBestDecision” in each node. But this is not convenient, particularly if the tree has numerous layers. Based on this concern, a mechanism was provided to allow users to indicate that they would like the software to display – for each decision node – the choice that was suggested to be the most favourable in light of model assumptions.

The time line mentioned as requirement (6) is to present the time interval associated with each successive layer of the tree. For health-oriented simulation models, the times of System Dynamics models are frequently mapped onto calendar dates and times. For compatibility with other tools (e.g., Excel), sometimes the Vensim timeline makes use of the date encoding scheme drawn from Excel; within this scheme 1 means “1900-01-01”. To facilitate reading and support of models where the unit of time is not days, it is important to allow for showing the timeline using calendar days or using simple real numbers. Figure 5.5 shows the effect of changing the time line to date. It can be recognized from the diagram that the interval of the time separating the rootNode and the next decision node layer is 2 days. Figure 5.5 also shows there are several modes of presentation to choose (e.g. The system is in a mode where paths are shown to the selected node).

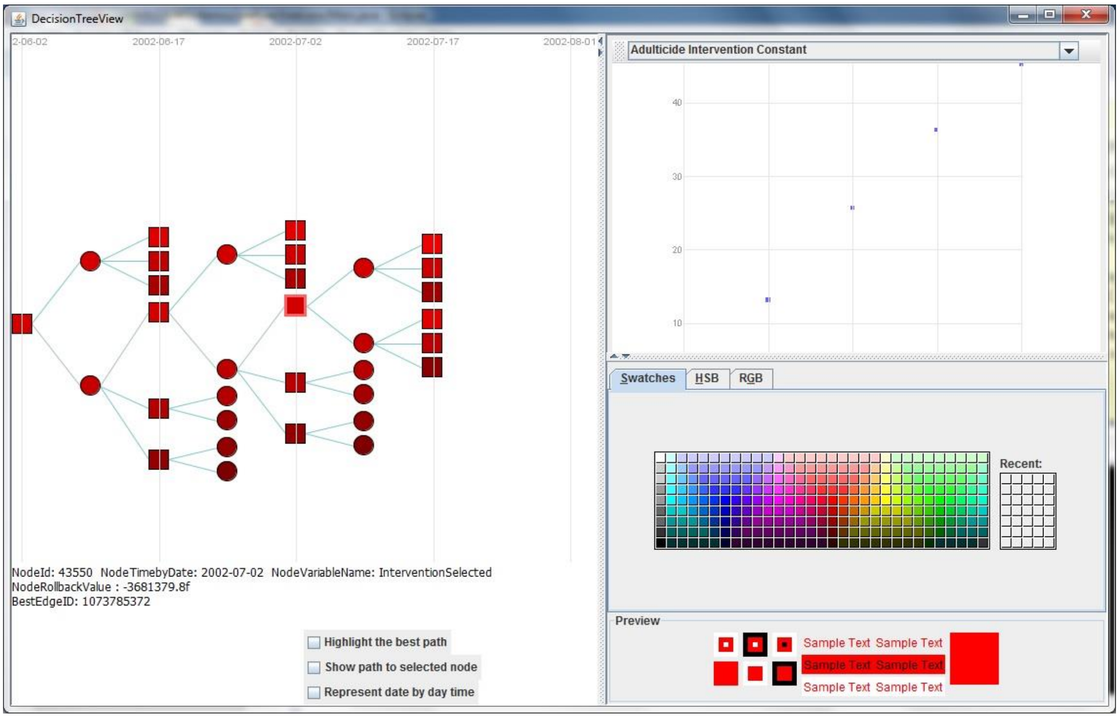

Figure 5.5: mouse over using label to present information for each tree element

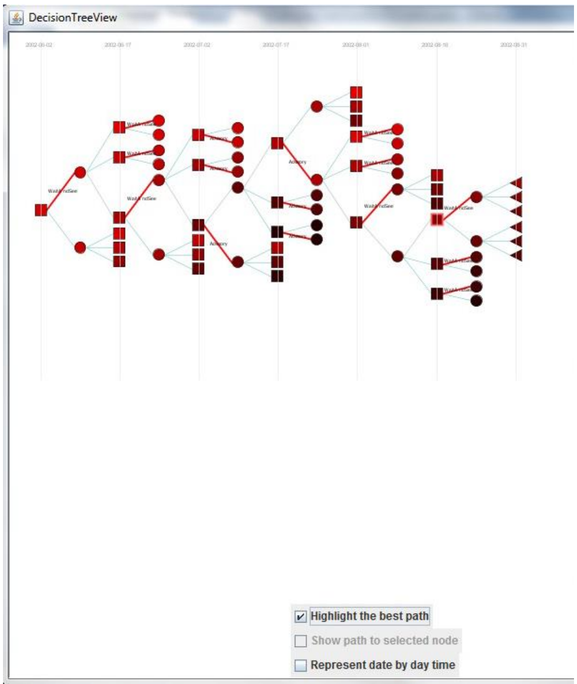

Figure 5.6: Highlighted tree for preferred decision

Each scenario (extending from the root node to a specific terminal node) is associated with specific trajectories for each simulation model variable. Given that users are frequently interested in understanding the outcomes associated with particular paths of decisions and events over time, a facility was provided to allow users to display this information. To satisfy objective (7), the following feature was introduced: when users click on a Terminal Node, the system treats them as having selected the analysis of a specific scenario. A table needs to display the specific parameter's value based on the Decision Tree's time line.

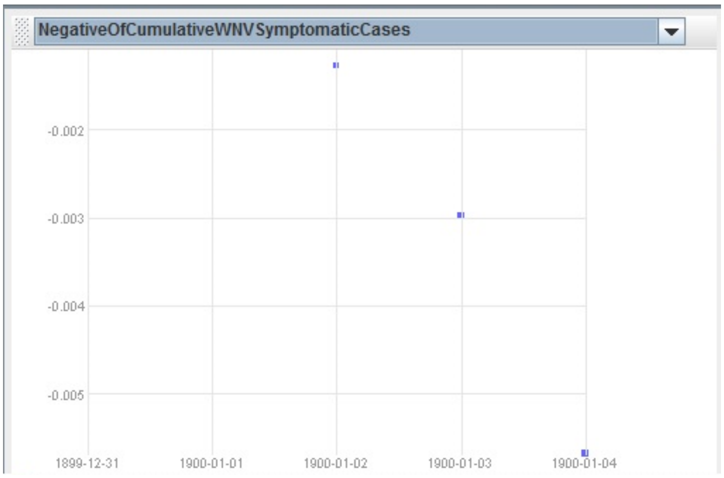

Figure 5.7: Timeplot for model variables in the current scenario

The drop down box at the top of the dialog box presents the variables in the Vensim model associated with the tree. When one of the variable names has been selected by the dropdown box (see Figure 5.7), the variable's value in the current scenario is presented in the time plot, such as that shown in Figure 5.7. As Figure 5.8 shows, when a terminal node is chosen, the whole path's information from root node to the current terminal node would is presented, including the rollback value for each node and the edge's name along this path (see Figure 5.8).

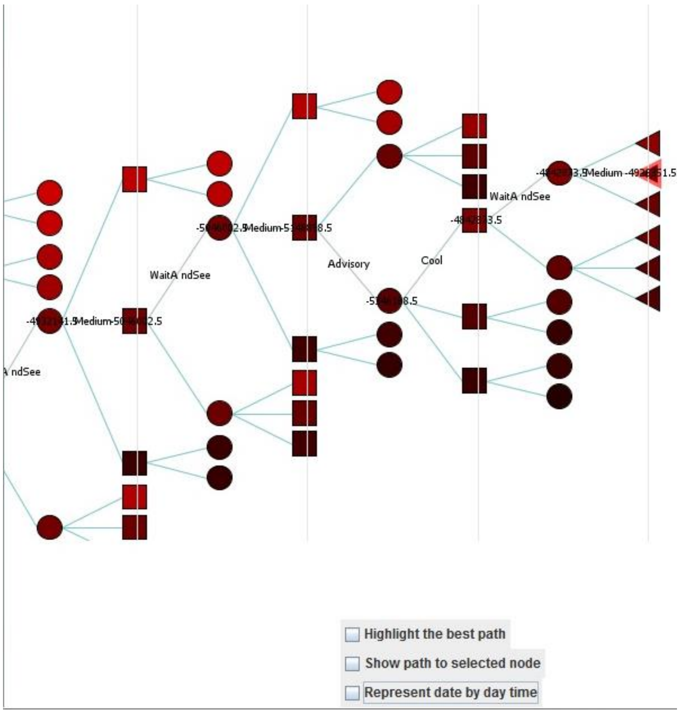

Figure 5.8: Specific path from root to terminal node

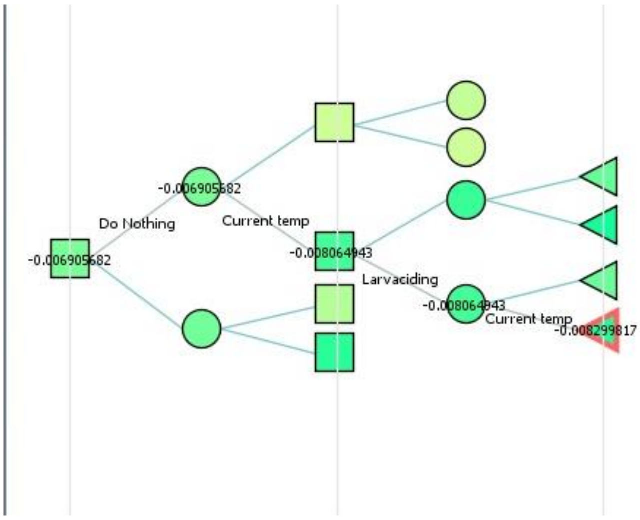

Figure 5.9: The information for the entire scenario

This visualization toolkit is designed for use by a wide variety of users. Following discussion with one particular user (West Nile virus researcher and Professor Tasha Epp of the Western College of Veterinary Medicine), both to address aesthetic preferences and because of color blindness and other reasons, it is valuable to provide the user with the capacity to easily change colours in this toolkit. We provide 3 approaches for choosing colors.

a. Swatches

The system allows Users to change color by the color panel. Recently chosen colors will be listed in the "Recent" color table. The table also provides the preview of what this color will look like when used in the tree.

b. HSB

The HSB color specification interface has three sliders to let users choose the current color's hue, saturation and brightness. Modify those parameters allows color to be changed in a more subtle fashion.

c. RGB

Color can further be specified with a specification of Red, Green and Blue components. Once we change one of those, the color will be changed. Similar to HSB, the RGB mechanism also provides three sliders for users to change Red, Green and Blue content in the color.

5.4. Discussion

The visualization mechanisms described in this chapter provide flexible and intuitively presented details when interacting with a Decision Tree. The software can provide an entire tree view to give a policymaker a general view of this tree, or information on particular scenarios. The color of a given node can communicate the magnitude of the rollback value for that node. To obtain details for each leaf or branch, there are several options to choose: Users can represent the specific scenario from root node, highlight the preferred decision of the tree selected given model assumptions, and also give the entire path information from root node to terminal. To help future research, it also can retrieve the time trajectory of variables in the simulation model required for understanding each different scenario in the decision tree.

This toolkit greatly improved the capacity of the user to interact with the decision tree, compared with the XML file. It gives an idea of data and visualization interaction based on the XML file. By employing the “Prefuse” API to construct a toolkit, we can easily customize the functions that users require.

Chapter 6

Conclusions

6.1. Summary

In this thesis, we refined Decision Tree software that integrates System Dynamics simulation and decision analysis, so that we can evaluate adaptive policy decisions over time in situations where the choice of the best decision at a given point depends not only on past decisions but also on observed and future unknown sequences of chance events. We further extended, refined and enhanced the accuracy of the WNv model construct by Yee [17]. We compared the previous version and the current version to evaluate the improvement in terms of matching historic data. In addition, by importing the WNv model into AnyLogic, we created patches for a spatially explicit model, which was further provided with a user interface to monitor the interaction between patches and their neighbours. Finally, we created a toolkit to visually represent and manipulate a decision tree structure and results.

6.2. Deliverables of the research work

The deliverables of the thesis includes:

- Decision Tree analysis software with an easy to use graphic user interface. This software offers 4 major types of functionality:
    - Build a tree that can easily exported as an XML file.
    - Run a tree together with a System Dynamics model.
    - Interactively view the tree and see the relevant information of each scenario.
    - Examine the results of the backward induction to determine the preferred decision.
- A WNv model extending and refining on Karen Yee's original version. In addition to new structure and parameter, this involves resolved defects in the core tab and calibration tab.
- An Agent-based model which includes the WNv model in each patch, and representation of interactions between patches.

6.3. Future work

6.4.1 Decision Tree software

In order to improve system robustness and utility, the following steps remain to be addressed.

If the users want to design a Decision Tree by employing the "create Tree" interface, there are a variety of restrictions that restrict the functionality offered. Firstly, the tree currently must be a Uniform tree with the same number of branches for every decision node; a similar relationship holds for each event node. In the decision tree created, event layers and decision layers are required to alternate with each other. For example, if the nodes in the first layer are decision nodes, the second layer must be an event layer and so on. However, in a real environment, this may be too restrictive.

6.4.2 The system dynamic model of WNv

We are encouraged by the results of this preliminary study. One can improve the model further with the following steps:

1. Recent literature suggests that the American Robin is responsible for the majority of WNv-infected mosquitoes[41]. One can plan to test the sensitivity of the optimised WNv model [1] with data from the American Robin. If the model responds we will substitute American Crow data with American Robin data to simulate disease transmission among mosquitoes, American Robins, and humans.
2. The values obtained from literature in the current WNv model [1] are five years old. We will apply newer data sets to update the model's accuracy.

References

[1] Sterman, John D. (2001). System Dynamics Modeling. *California management review, 43*(4), 8-28.

[2] MIT System Dynamics in Education Project (SDEP). Retrieved date: February 15, 2012, from: http://web.mit.edu/sysdyn/sdep.html.

[3] Lemon, Stephenie C, Roy, Jason, Clark, Melissa A, Friedmann, Peter D, & Rakowski, William. (2003). Classification and regression tree analysis in public health: methodological review and comparison with logistic regression. *Annals of Behavioral Medicine, 26*(3), 172-181.

[4] Yee, K. Osgood, N., An, W. (2011). A hybrid system dynamics and decision analysis tool for effective strategy selection to control for a West Nile virus epidemic. *Extended abstract and presentation at System Dynamics Winter Conference 2011. Invited contributions only*(Austin, Texas.), 2pp.

[5] Petersen, Lyle R, & Marfin, Anthony A. (2002). West Nile virus: a primer for the clinician. *Annals of Internal Medicine, 137*(3), 173-179.

[6] Komar, Nicholas. (2003). West Nile virus: epidemiology and ecology in North America. *Advances in virus research, 61*, 185-234.

[7] Ludwig, George V, Calle, Paul P, Mangiafico, Joseph A, etc., & McNamara, Tracey. (2002). An outbreak of West Nile virus in a New York City captive wildlife population. *American Journal of Tropical Medicine and Hygiene, 67*(1), 67.

[8] Osgood, N. Dynamic Decision Problems: Hybrid Use of Decision Trees & System Dynamics Models. (2013).

[9] Wonham, Marjorie J, Lewis, Mark A, Rencławowicz, Joanna, & Van den Driessche, P. (2006). Transmission assumptions generate conflicting predictions in host–vector disease models: a case study in West Nile virus. *Ecology Letters, 9*(6), 706-725.

[10] Cruz-Pacheco, Gustavo, Esteva, Lourdes, Montaõ-Hirose, Juan Antonio, & Vargas, Cristobal. (2005). Modelling the dynamics of West Nile virus. *Bulletin of mathematical biology, 67*(6), 1157-1172.

[11] Osgood, N. (2005). *Combining System Dynamics and Decision Analysis for Rapid Strategy Selection.* Paper presented at the Proceedings of the 23rd International Conference of System Dynamics Society.

[12] Pennsylvania State University. West Nile Encephalitis in Horses.(2008).

[13] Reisen, William K, Ying, F, & Martinez, Vincent M. (2006). Effects of temperature on the transmission of West Nile virus by Culex tarsalis (Diptera: Culicidae). *Journal of medical entomology, 43*(2), 309-317.

[14] Brailsford, SC, Berchi, R, De Angelis, V, & Mecoli, M. (2007). *System Dynamics models to assess the risk of mosquito-borne diseases and to evaluate control policies.* Paper presented at the International Conference of the System Dynamics Society.

[15] Osgood N., Yee K., An W., Grassmann W. (2013). Addressing Dynamically Complex Decision Problems Using Decision Analysis and Simulation

[16] Leung, T. The designer and developer of the prototype of the Decision Tree software in CMPT 405 who was in Professor Nathaniel Osgood's course. (2008).

[17] Yee, K., Osgood,N. The WNv System Dynamic model prototype (Figure 3.3)is constructed by Karen Yee who was Research Associate in the Laboratory for Computational Epidemiology and Public Health Informatics

[18] Raiffa, H, & Schlaifer, R. (1961). Applied Statistical Decision Theory (Harvard Business School Publications).

[19] Howard, Ronald A. (1966). *Decision analysis: Applied decision theory*: Stanford Research Institute.

[20] De Ville, B. (2006). *Decision trees for business intelligence and data mining: using SAS enterprise miner*: SAS Institute.

[21] Kiberd, Bryce A, & Forward, K. (2004). Screening for West Nile virus in organ transplantation: a medical decision analysis. *American Journal of Transplantation, 4*(8), 1296-1301.

[22] Zohrabian, A, Hayes, Edward B, & Petersen, Lyle R. (2006). Cost-effectiveness of West Nile virus vaccination. *Emerging infectious diseases*.

[23] Overview of AnyLogic. Retrieved date: April 14,2014,from: http://www.anylogic.com/overview.

[24] Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America, 99*(Suppl 3), 7280-7287.

[25] Eberlein, RL, & Peterson, DW. (1994). Understanding Models with Vensim, Modeling for Learning Organizations: Productivity P professor Nathaniel Osgood’s course. (2009).ress, Portland, OR.

[26] Kirkwood, Craig W. (1998). System Dynamics Methods: A Quick Introduction, 1998 http://www.public.asu.edu/~kirkwood/sysdyn/SDIntro/SDIntro.htm.

[27] Sterman, John D. (2000). *Business dynamics: systems thinking and modeling for a complex world* (Vol. 19): Irwin/McGraw-Hill Boston.

[28] Mecoli, M, De Angelis, V, & Brailsford, Sally C. (2010). *Modelling the risk of mosquito-borne diseases by System Dynamics: The case of human travel between different geographic regions*. Paper presented at the Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on.

[29] Gao, A. 2006. "A Hybrid Framework for Selecting Optimal Policy for Complex System". In Proceedings, CMPT 880 Research Symposium, April 2006. Department of Computer Science, University of Saskatchewan, Saskatoon, Canada."

[30] Koutsofios, E, & North, S. (1991). Drawing graphs with dot: Technical Report 910904-59113-08TM, AT&T Bell Laboratories, Murray Hill, NJ.

[31] Ventana Systems, Inc,” Vensim Reference Manual”. Retrieved date: August 22, 2010, 210-211, from: http://www.vensim.com/documentation.html.

[32] Unnasch, Robert S, Sprenger, T, Katholi, Charles R, Cupp, Eddie W, Hill, Geoffrey E, & Unnasch, Thomas R. (2006). A dynamic transmission model of eastern equine encephalitis virus. *Ecological modelling, 192*(3), 425-440.

[33] Ludwig, A, Bigras-Poulin, M, Michel, P, & Bélanger, D. (2010). Risk factors associated with West Nile virus mortality in American Crow populations in Southern Quebec. *Journal of wildlife diseases, 46*(1), 195-208.

[34]Reisen,W.,Barker,C.,Carney,R.,Lothrop,H.,Wheeler,S.,JenniferL.,Madon,M.,Takahashi,R.,Carroll, B.,Garcia,S.,Fang,Y.,Shafii,M.,Kahl,N.,Ashtari,S.,Kramer,V.,Glaser,C.,Jean,C. (2006). Role

of corvids in epidemiology of West Nile virus in southern California. Journal of medical entomology, 43(2), 356-367.

[35] Foppa, Ivo M, & Spielman, Andrew. (2007). *Does reservoir host mortality enhance transmission of West Nile virus?* (Vol. 4).

[36] Culex tarsalis Mosquitoes Arrive in the City. Retrieved date: April 14, 2014, from: http://www.saskatoon.ca/FORUM/News%20Releases/Pages/CulextarsalisMosquitoesArriveintheCity.aspx.

[37] Osgood, N. Computational Epidemiology and Public Health Informatics Laboratory technical note

[38] Wikipedia explanation of DOT. Retrieved date: April 14,2014, from http://en.wikipedia.org/wiki/DOT_(graph_description_language).

[39] Introduction of Processing. Retrieved date: April 14,2014, from http://processing.org/.

[40] Introduction of the prefuse visulization toolkit. Retrieved date: April 14,2014, from http://prefuse.org/.

[41] Kilpatrick, A Marm, Daszak, Peter, Jones, Matthew J, Marra, Peter P, & Kramer, Laura D. (2006). Host heterogeneity dominates West Nile virus transmission. *Proceedings of the Royal Society B: Biological Sciences*, *273*(1599), 2327-2333.

Appendix A

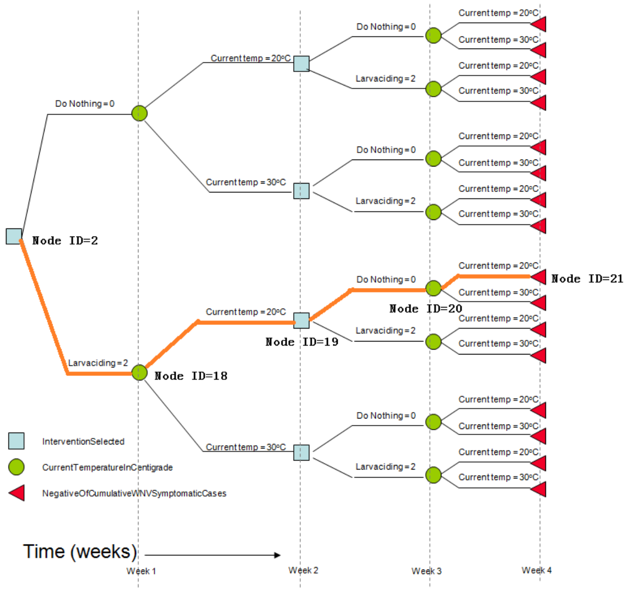

Figure A: Representation of a specific decision rule

The path is presented in Figure .

1. Set game time interval in Vensim

   The Time interval means the time from the current layer to the next layer. The command used in the prototype and adapted in our code is “"GAME>GAMEINTERVAL|"+`deltaTime`”. The variable deltaTime is the interval of time for each layer. In our case, the time interval in the decision layer and event layers are 1.0 as shown in Figure 3.5, except the time interval at the root node is 0.0. The time interval is very significant in simulation.

2. Set the variable value in Vensim

   For each scenario, use the command “"SIMULATE>`SETVAL`|" + `varName` + " = " + `value`” to set the variable value in the Vensim model. In our example path (orange path in Figure 9), at the root node, set the value of “`InterventionSelected`” equal to 2. In the first event node (Node ID=18), set the value of “`CurrentTemperatureInCentigrade`” equal to 20. At the second decision node (Node ID=19) the decision is implemented by setting `InterventionSelected` to 0Similarly, the second event node (Node ID=20) is implemented by setting `CurrentTemperatureInCentigrade` to 20. Those values are all set in the Vensim model in Game mode when running the simulation forward towards the terminal node.

3. Advance the model in Game mode

   To run the model forward, we use the command “GAME>`GAMEON`”. This command indicates to simulate the model step by step. Each step to run the simulate advances the model time by the amount specified by the time interval set in step 1.

After the three steps shown above have been performed recursively (applied throughout the tree), we can save the output from the system, a file in XML format. In the original prototype, several attributes of the parsed tree were be presented in the XML file, but that information omitted some important properties. I added the "`rollBackValue`" attribute for decision and chance (event) nodes, "`nameOfBestDecision`", "`idOfBestDecision`" attribute for each decision node, and an "edgeName" attribute for each edge. Saving away such information in a file is very important for analysis.

Appendix B

Variable list in Figure 3.3 (From the top to the bottom)

Top fragment:

Number of adult blood meals per day for Given Temperature
Number of adult blood meals per day
Egg Laying by Adult Female Mosquitoes Density
Egg Density
Birth of Mosquito Larvae
CurrentTemperatureInCentigrade
Virus Incubation Rate
Virus Incubation Threshold Temperature

Middle fragment:

InterventionSelected
Vaccine Rate
Vaccine
Entrants of Humans
Susceptible Humans
"Newly Infected Pre-symptomatic Human Cases"
Deaths from Susecptible Humans
Loss of Immunity of Recovered and WNV Immune Patients
Mean Time to Waning Immunity

Bottom fragment:

"Progression to Non-Hospitalized WNF"
Hospitalized for WNF
New Symptomatic Cases
Hospitalized for meningitis
Hospitalized for paralysis
Hospitalized for M and E
CumulativeWNVSymptomaticCases
NegativeOfCumulativeWNVSymptomaticCases